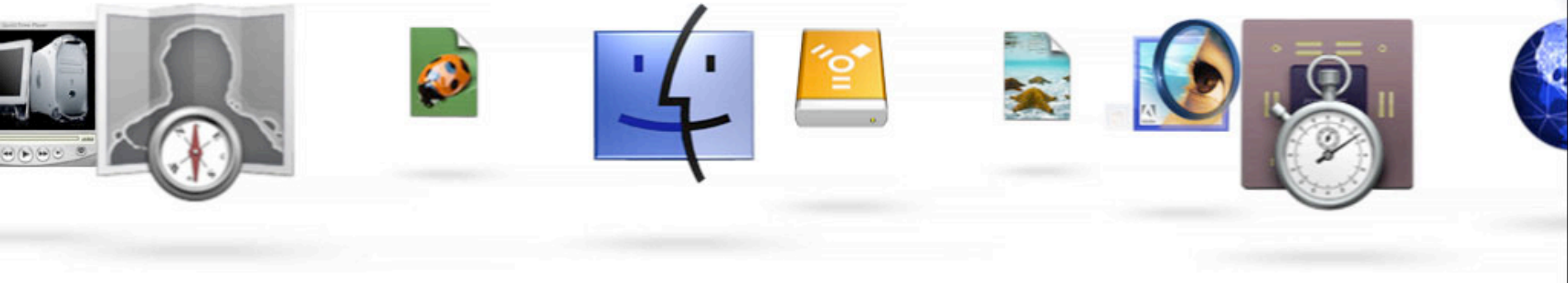




Accessibility and Carbon

Session 209





Accessibility and Carbon

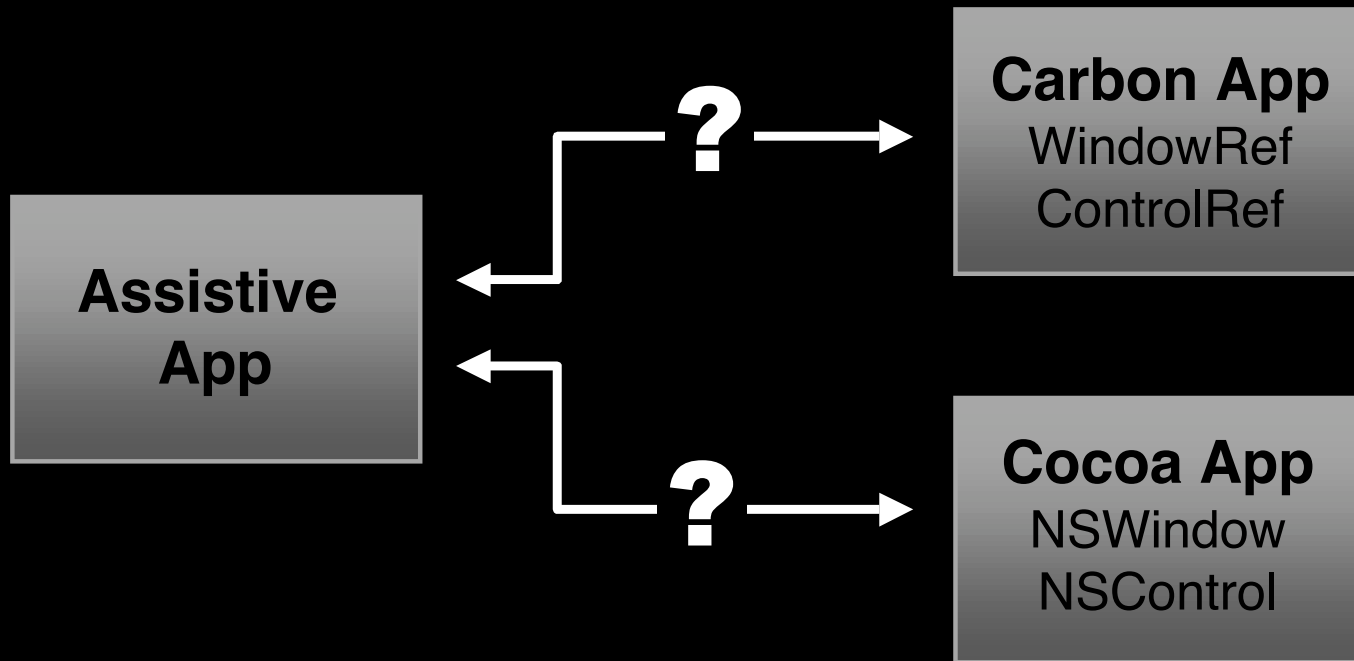
Guy Fullerton
HI Toolbox Engineer and
Year-Round Shorts Wearer

Agenda

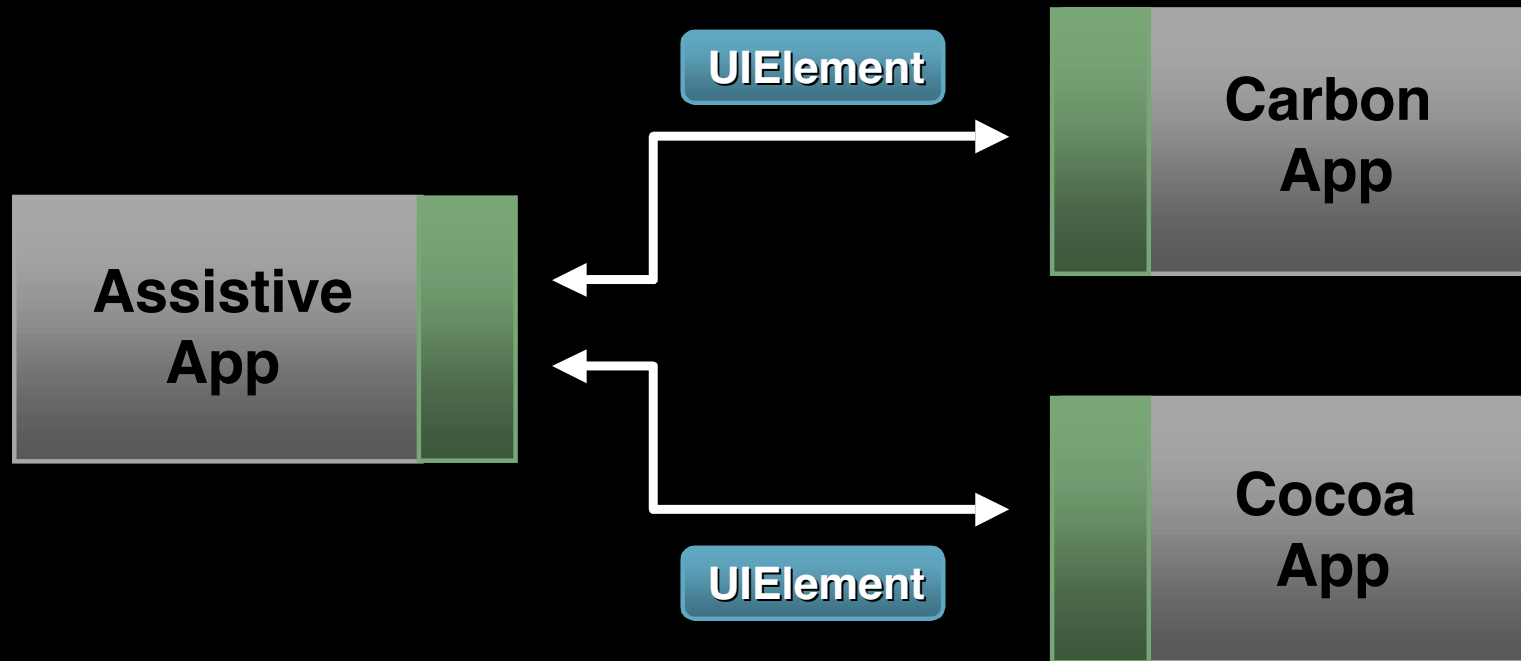
- Accessibility Carbon Events
- Carbon UIElements
- HICommands
- Notifications
- Keyboard Navigation



UI Representation—Problem



UI Representation—Solution



UI Representation—Solution



Making an App Accessible

- Describe interface to Assistive App
 - Find UIElements at a screen point
 - Traverse hierarchy of UIElements
- Allow Assistive Apps to invoke functionality
- Inform Assistive Apps of changes in your App



Accessibility Carbon Events

- Correspond to the Accessibility C APIs
- Three basic types of Accessibility Carbon Events



Accessibility Carbon Events

- Find UIElements in your interface
 - kEventAccessibleGetChildAtPoint
 - kEventAccessibleGetFocusedChild
 - kEventAccessibleGetNamedAttribute
 - When asking for “AXChildren” or “AXParent”



Accessibility Carbon Events

- Gather information about UIElements
 - kEventAccessibleGetAllAttributeNames
 - kEventAccessibleGetNamedAttribute
 - kEventAccessibleIsNamedAttributeSettable
 - kEventAccessibleGetAllActionNames
 - kEventAccessibleGetNamedActionDescription



Accessibility Carbon Events

- Manipulate UIElements
 - kEventAccessibleSetNamedAttribute
 - kEventAccessiblePerformNamedAction



Sidebar: Carbon Events?!

- Can be adopted piecemeal
- Work with `WaitNextEvent`
- Easy to add to existing code
- Session 203: Migrating to Carbon Events



Accessibility Carbon Events

- Sent directly to HIObjects in your app
 - WindowRefs
 - ControlRefs
 - MenuRefs
 - Other HIObject subclasses
- System HIObjects are already accessible!
- Session 204 explains HIObjects



Makes Sense So Far . . .

- What are a window's children?
- What is a push button's title?
- What is a menu's size?
- What is a check box's value?



But What About . . .

- Menu items?
 - What is a menu item's title?
- Data Browser cells?
 - What is a cell's position?
- Scroll Bar arrow?
 - What is the arrow's action description?
- You cannot send Carbon Events to subcomponents



The HIObject Does the Work

- Accessibility Carbon Events for subcomponents are serviced by the “owning” HIObject
- MenuRef handles the menu items
- Data Browser ControlRef handles the cells
- Scroll Bar ControlRef handles the arrows



Representation Must Be Flexible

- Whole Windows, Controls, or Menus
- Subcomponents of Windows, Controls, or Menus
- Totally custom parts of apps



AXUIElementRef

- Otherwise known as a UIElement
- Represents any Accessible widget on the screen
- It is a CTypeRef



UIElement Is a Proxy

UIElement

Process ID: 0x1942 (“MyApp”)

**Data: <Framework-specific
References>**



UIElement Is a Proxy



UIElements for Carbon Apps

- Framework-specific references:
 - HIObjectRef
 - 64-bit Identifier
- AXUIElementCreateWithHIObjectAndIdentifier
- AXUIElementGetHIObject
- AXUIElementGetIdentifier



UIElement's HIObject

- Defines the recipient of the Carbon Events pertaining to the UIElement
- Usually a Control, Window, or Menu
- Can be something else



UIElement's 64-bit Identifier

- One HIObject can represent many UIElements
- Zero implies the whole HIObject
- Non-zero means a subcomponent of an HIObject
- 64-bit space is private to the HIObject!



MenuRef & 0 → Menu

UIElement

Process ID: 0x1942

Data: MenuRef: 0x26F4,
Identifier: 0

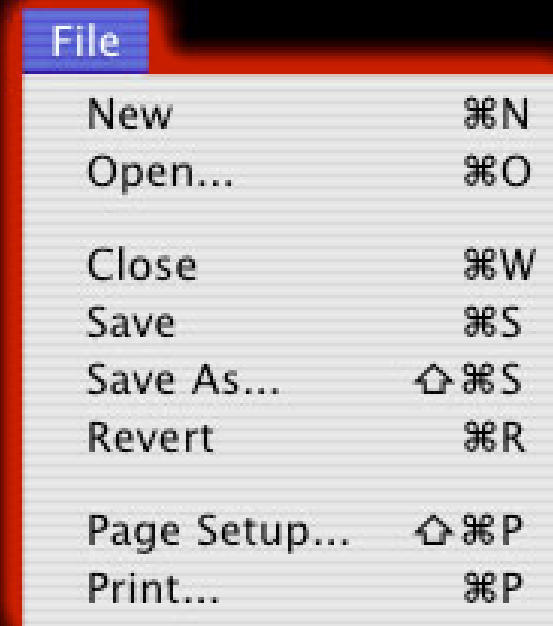


MenuRef & 0 → Menu

UIElement

Process ID: 0x1942

Data: MenuRef: 0x26F4,
Identifier: 0

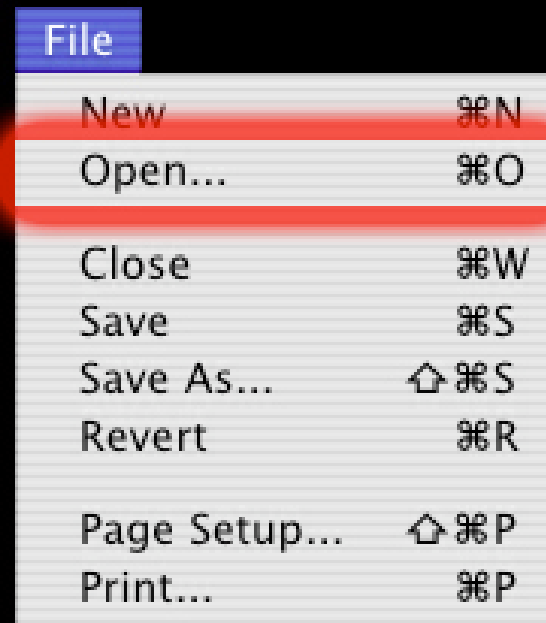


MenuRef & 2 → Menu Item

UIElement

Process ID: 0x1942

Data: MenuRef: 0x26F4,
Identifier: 2

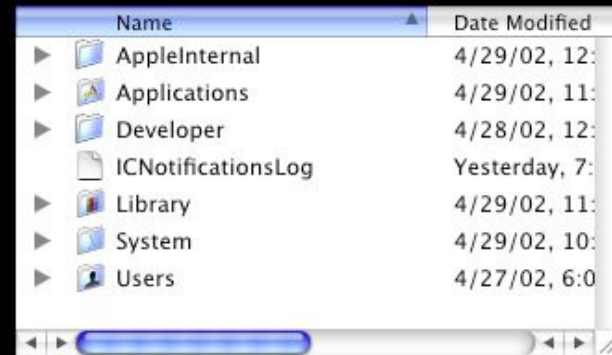


ControlRef & 0 → Outline

UIElement

Process ID: 0x1942

Data: ControlRef: 0x4E56,
Identifier: 0



A screenshot of a file explorer window showing a list of folders and files. The window has a title bar and a scroll bar at the bottom. The list is organized into two columns: 'Name' and 'Date Modified'. The folders listed are AppleInternal, Applications, Developer, Library, System, and Users. The files listed are ICNotificationsLog and a file named 'Yesterday, 7:'. The dates and times are listed in the 'Date Modified' column.

Name	Date Modified
▶ AppleInternal	4/29/02, 12:
▶ Applications	4/29/02, 11:
▶ Developer	4/28/02, 12:
ICNotificationsLog	Yesterday, 7:
▶ Library	4/29/02, 11:
▶ System	4/29/02, 10:
▶ Users	4/27/02, 6:0

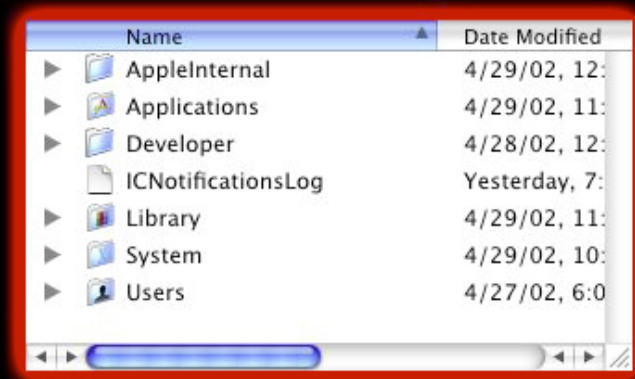


ControlRef & 0 → Outline

UIElement

Process ID: 0x1942

Data: ControlRef: 0x4E56,
Identifier: 0

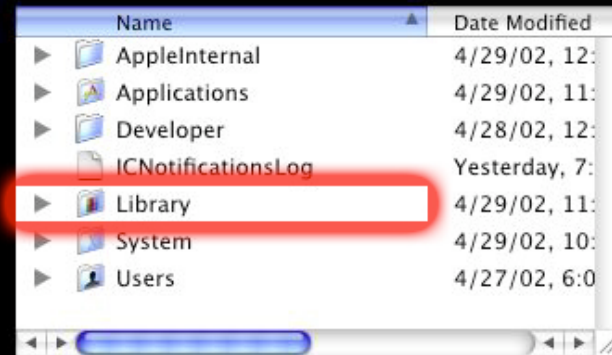


ControlRef & 0x1234 → Cell

UIElement

Process ID: 0x1942

Data: ControlRef: 0x4E56,
Identifier: 0x1234



Name	Date Modified
▶ AppleInternal	4/29/02, 12:
▶ Applications	4/29/02, 11:
▶ Developer	4/28/02, 12:
▶ ICNotificationsLog	Yesterday, 7:
▶ Library	4/29/02, 11:
▶ System	4/29/02, 10:
▶ Users	4/27/02, 6:0

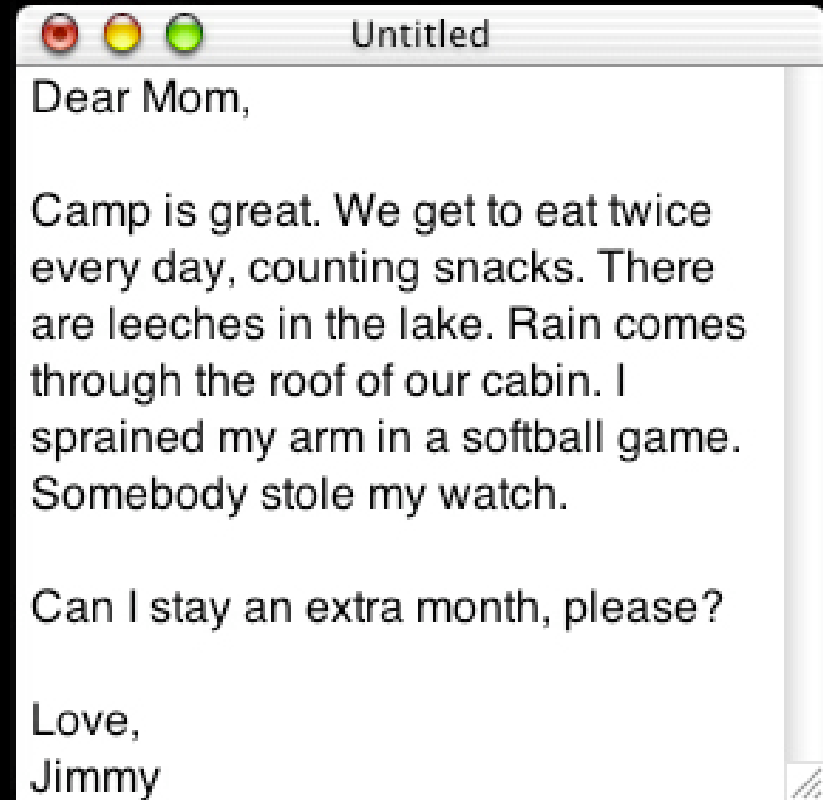


HIObjectRef & 0 → Document

UIElement

Process ID: 0x1942

Data: HIObjectRef: 0x03DA,
Identifier: 0

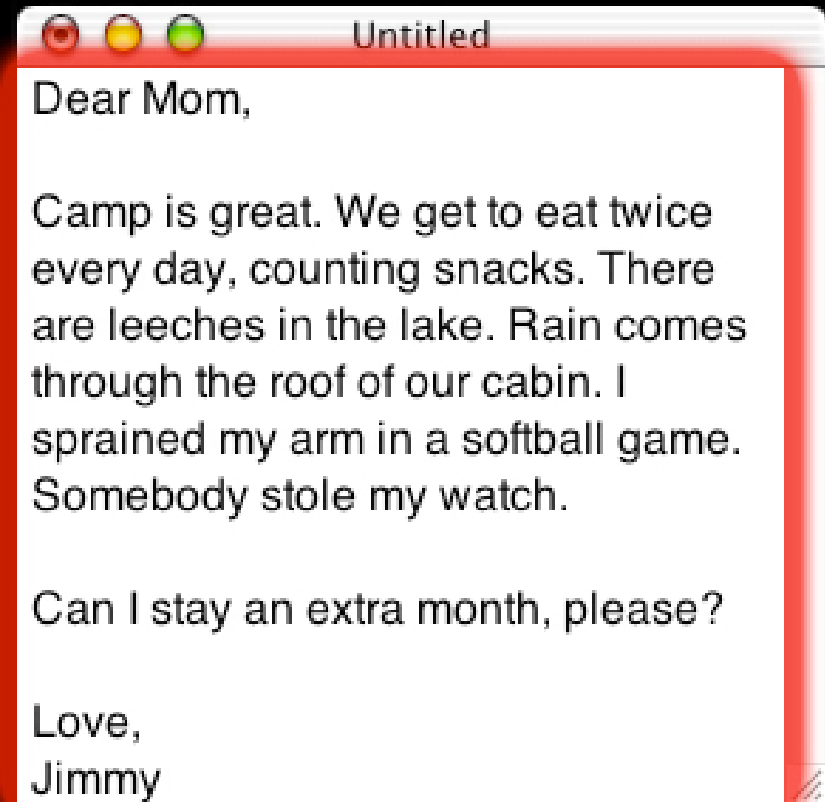


HIObjecRef & 0 → Document

UIElement

Process ID: 0x1942

Data: HIObjecRef: 0x03DA,
Identifier: 0

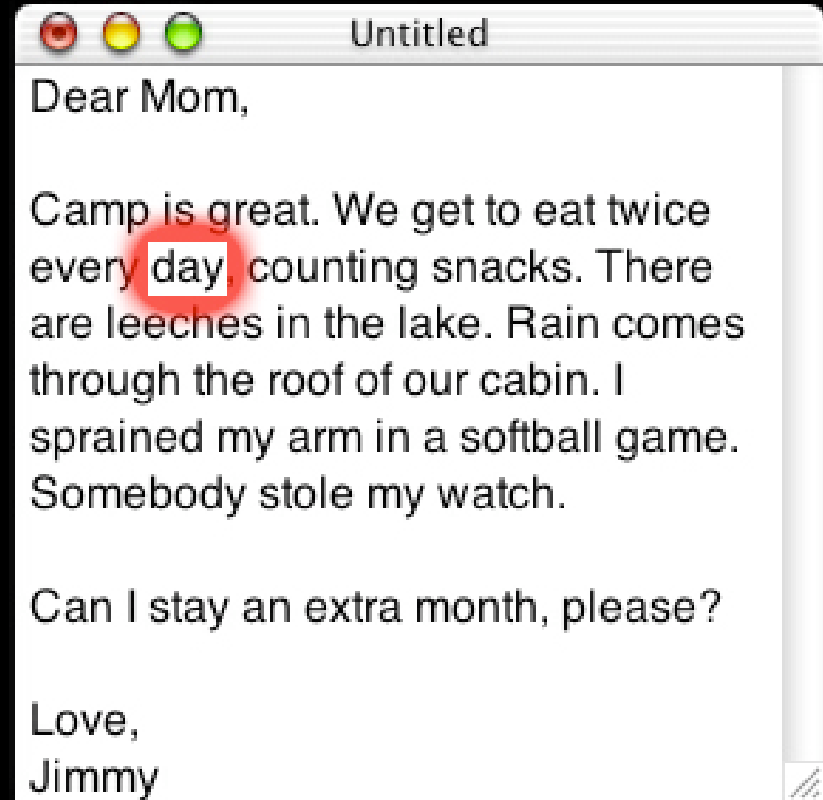


HIObjectRef & 12 → 12th Word

UIElement

Process ID: 0x1942

Data: HIObjectRef: 0x03DA,
Identifier: 12



Representing Your Application

- UIElements form a containment hierarchy
- System already implements much of it
 - Application contains Windows
 - Windows contain Controls
 - Controls contain more Controls



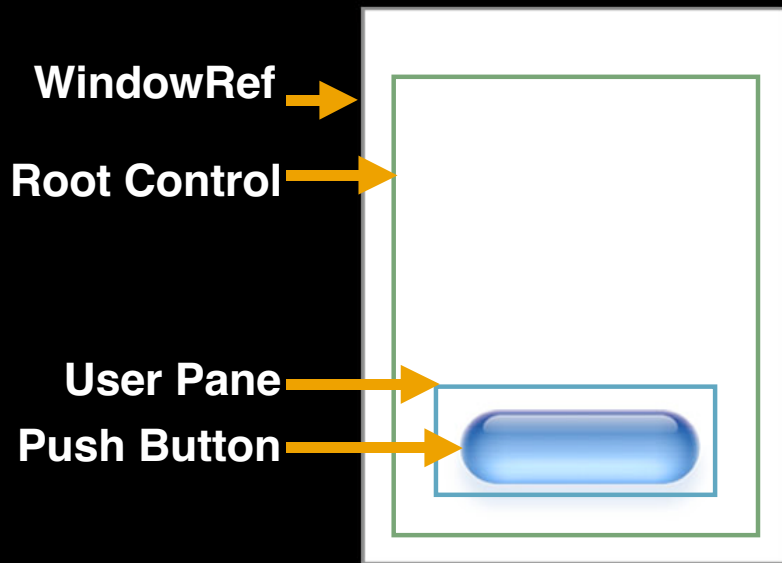
Representing Your Application

- Work is necessary to represent other things
 - PowerPlant Views
 - Text engines
- Create HIObjects
 - Wrap your implementation objects
- Override default handlers
 - Inject your UIElements into the hierarchy

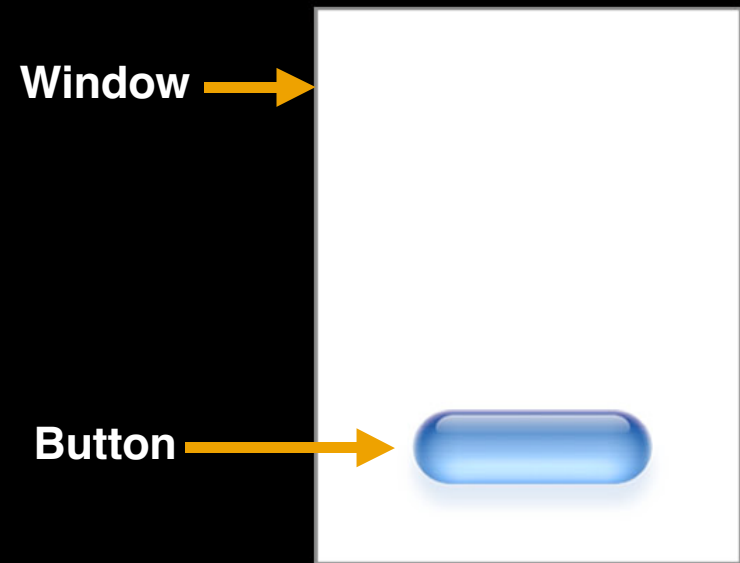


Sidebar: Ignored UIElements

Actual Hierarchy



Accessibility Hierarchy



Sidebar: Ignored UIElements

- Carbon does the hard work automatically
- Set at the HIObjectRef level:
 - HIObjectSetAccessibilityIgnored



Producing UIElements



Producing UIElements

- By position
 - kEventAccessibleGetChildAtPoint
- Via the focus chain
 - kEventAccessibleGetFocusedChild
- Through familial references
 - kEventAccessibleGetNamedAttribute
 - When asking for “AXChildren” or “AXParent”



By Position

- kEventAccessibleGetChildAtPoint
 - UIElement being queried
 - Global screen point
- Pass back the *immediate* child UIElement at the point
- eventNotHandledErr → no child at the point



Example: Get Child At Point

- I have a text editing engine
- Its UIElement should be a child of the Window
- I have installed a handler on the Window
 - kEventAccessibleGetChildAtPoint
 - Splice in the text engine as a child of the Window



Example: Get Child At Point

```
GetEventParameter( event,  
    kEventParamAccessibleObject, typeCFTTypeRef,  
    NULL, sizeof( windowElement ), NULL,  
    &windowElement );
```

```
AXUIElementGetIdentifier( windowElement, &id );
```

```
if ( id != 0 )  
    return eventNotHandledErr;
```



Example: Get Child At Point

```
GetEventParameter( event,  
    kEventParamAccessibleObject, typeCFTTypeRef,  
    NULL, sizeof( windowElement ), NULL,  
    &windowElement );
```

```
AXUIElementGetIdentifier( windowElement, &id );
```

```
if ( id != 0 )  
    return eventNotHandledErr;
```



Example: Get Child At Point

```
GetEventParameter( event,  
    kEventParamAccessibleObject, typeCFTTypeRef,  
    NULL, sizeof( windowElement ), NULL,  
    &windowElement );
```

```
AXUIElementGetIdentifier( windowElement, &id );
```

```
if ( id != 0 )  
    return eventNotHandledErr;
```



Example: Get Child At Point

```
GetEventParameter( event,  
    kEventParamMouseLocation, typeHIPoint,  
    NULL, sizeof( point ), NULL, &point );
```

```
// ... convert point from global to local ...
```

```
if ( ! MyTextEditorContainsPoint( myEditor, &point ) )  
    return eventNotHandledErr;
```



Example: Get Child At Point

```
GetEventParameter( event,  
                  kEventParamMouseLocation, typeHIPoint,  
                  NULL, sizeof( point ), NULL, &point );
```

```
// ... convert point from global to local ...
```

```
if ( ! MyTextEditorContainsPoint( myEditor, &point ) )  
    return eventNotHandledErr;
```



Example: Get Child At Point

```
GetEventParameter( event,  
                  kEventParamMouseLocation, typeHIPoint,  
                  NULL, sizeof( point ), NULL, &point );
```

```
// ... convert point from global to local ...
```

```
if ( ! MyTextEditorContainsPoint( myEditor, &point ) )  
    return eventNotHandledErr;
```



Example: Get Child At Point

```
GetEventParameter( event,  
                  kEventParamMouseLocation, typeHIPoint,  
                  NULL, sizeof( point ), NULL, &point );
```

```
// ... convert point from global to local ...
```

```
if ( ! MyTextEditorContainsPoint( myEditor, &point ) )  
    return eventNotHandledErr;
```



Example: Get Child At Point

```
element = MyTextEditorCreateUIElement( myEditor );
```

```
SetEventParameter( event,  
    kEventParamAccessibleChild, typeCFTTypeRef,  
    sizeof( element ), &element );
```

```
CFRelease( element );
```

```
return noErr;
```



Example: Get Child At Point

```
element = MyTextEditorCreateUIElement( myEditor );
```

```
SetEventParameter( event,  
    kEventParamAccessibleChild, typeCFTTypeRef,  
    sizeof( element ), &element );
```

```
CFRelease( element );
```

```
return noErr;
```



Example: Get Child At Point

```
element = MyTextEditorCreateUIElement( myEditor );
```

```
SetEventParameter( event,  
    kEventParamAccessibleChild, typeCFTTypeRef,  
    sizeof( element ), &element );
```

```
CFRelease( element );
```

```
return noErr;
```



Example: Get Child At Point

```
element = MyTextEditorCreateUIElement( myEditor );
```

```
SetEventParameter( event,  
    kEventParamAccessibleChild, typeCFTTypeRef,  
    sizeof( element ), &element );
```

```
CFRelease( element );
```

```
return noErr;
```



Example: Get Child At Point

```
element = MyTextEditorCreateUIElement( myEditor );
```

```
SetEventParameter( event,  
    kEventParamAccessibleChild, typeCFTTypeRef,  
    sizeof( element ), &element );
```

```
CFRelease( element );
```

```
return noErr;
```



Via the Focus Chain

- kEventAccessibleGetFocusedChild
 - UIElement being queried
- Pass back the *immediate* child UIElement in the focus chain
- eventNotHandledErr → no focused child

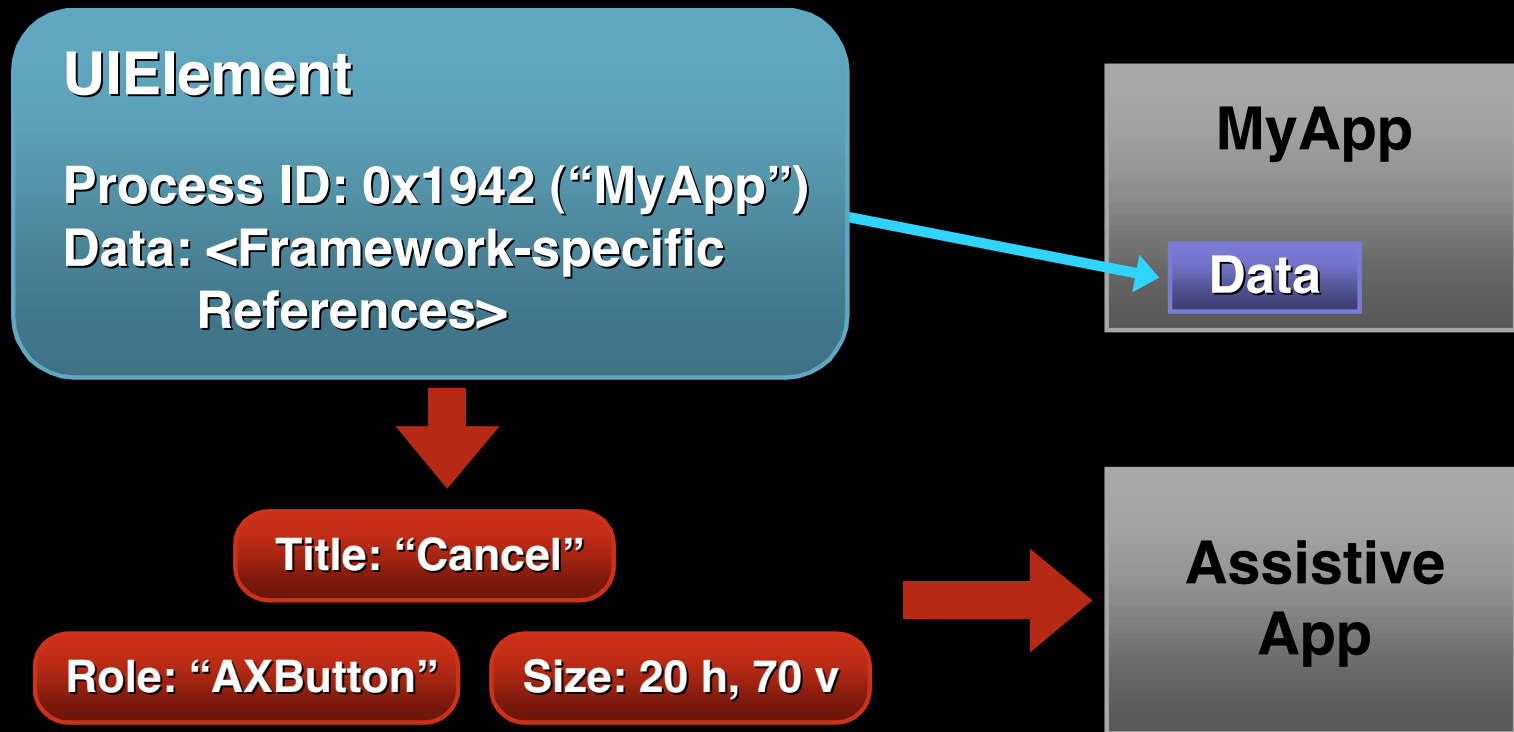


Through Familial References

- kEventAccessibleGetNamedAttribute
 - When asking for “AXChildren” or “AXParent”
- Just a specific case of Attribute fetching . . .



UIElement Attributes



UIElement Attributes

- Tagged data of a UIElement
- Examples
 - Children
 - Screen Position
 - Size
 - Selection of Edit Field
 - Focus of an Application



Required Attributes

- Role: CFStringRef
- Role Description: CFStringRef
- Size: AXSize (or HSize)
- Position: AXPosition (or HIPoint)
- Parent: AXUIElementRef
- Children: CFArrayRef of AXUIElementRefs



Get All Attribute Names

- `kEventAccessibilityGetAllAttributeNames`
 - UIElement being queried
 - `CFArrayRef` to modify
- Add all supported attribute names to `CFArrayRef`
- Many standard attribute names
 - `<HIServices/AXConstants.h>`



Example: All Attribute Names

```
GetEventParameter( event,  
                   kEventParamAccessibleAttributeNames,  
                   typeCFMutableArrayRef, NULL, sizeof( array ),  
                   NULL, &array );
```

```
CFArrayAppendItem( array, kAXRoleAttribute );  
// ... add the rest of the required attributes ...
```

```
CFArrayAppendItem( array, kAXTextAttribute );  
CFArrayAppendItem( array, kAXSelectionAttribute );
```

```
return noErr;
```



Example: All Attribute Names

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeNames,  
                  typeCFMutableArrayRef, NULL, sizeof( array ),  
                  NULL, &array );
```

```
CFArrayAppendItem( array, kAXRoleAttribute );  
// ... add the rest of the required attributes ...
```

```
CFArrayAppendItem( array, kAXTextAttribute );  
CFArrayAppendItem( array, kAXSelectionAttribute );
```

```
return noErr;
```



Example: All Attribute Names

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeNames,  
                  typeCFMutableArrayRef, NULL, sizeof( array ),  
                  NULL, &array );
```

```
CFArrayAppendItem( array, kAXRoleAttribute );  
// ... add the rest of the required attributes ...
```

```
CFArrayAppendItem( array, kAXTextAttribute );  
CFArrayAppendItem( array, kAXSelectionAttribute );
```

```
return noErr;
```



Get Named Attribute

- `kEventAccessibleGetNamedAttribute`
 - UIElement being queried
 - Attribute name `CFStringRef`
- Pass back the attribute's data
 - `kEventParamAccessibleAttributeValue`
 - Generally a `CTypeRef`



Example: Get Named Attribute

```
GetEventParameter( event,  
    kEventParamAccessibleAttributeName,  
    typeCFStringRef, NULL, sizeof( attribute ),  
    NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    // ... put text attribute data into the event ...  
}
```



Example: Get Named Attribute

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeName,  
                  typeCFStringRef, NULL, sizeof( attribute ),  
                  NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    // ... put text attribute data into the event ...  
}
```



Example: Get Named Attribute

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeName,  
                  typeCFStringRef, NULL, sizeof( attribute ),  
                  NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    // ... put text attribute data into the event ...  
}
```



Example: Get Named Attribute

```
string = MyTextEngineCreateString( engine );
```

```
SetEventParameter( event,  
    kEventParamAccessibleAttributeValue,  
    typeCFTypeRef, sizeof( string ), &string );
```

```
CFRelease( string );
```



Example: Get Named Attribute

```
string = MyTextEngineCreateString( engine );
```

```
SetEventParameter( event,  
    kEventParamAccessibleAttributeValue,  
    typeCFTyperef, sizeof( string ), &string );
```

```
CFRelease( string );
```



Example: Get Named Attribute

```
string = MyTextEngineCreateString( engine );
```

```
SetEventParameter( event,  
    kEventParamAccessibleAttributeValue,  
    typeCFTypeRef, sizeof( string ), &string );
```

```
CFRelease( string );
```



Is Named Attribute Settable?

- kEventAccessibleIsNamedAttributeSettable
 - UIElement being queried
 - Attribute name CFStringRef
- Pass back a Boolean
 - kEventParamAccessibleAttributeSettable
- eventNotHandledErr → not settable



Example: Is Attribute Settable?

```
GetEventParameter( event,  
    kEventParamAccessibleAttributeName,  
    typeCFStringRef, NULL, sizeof( attribute ),  
    NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    Boolean settable = true;  
    SetEventParameter( event,  
        kEventParamAccessibleAttributeSettable,  
        typeBoolean, sizeof( settable ), &settable );  
}
```



Example: Is Attribute Settable?

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeName,  
                  typeCFStringRef, NULL, sizeof( attribute ),  
                  NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    Boolean settable = true;  
    SetEventParameter( event,  
                      kEventParamAccessibleAttributeSettable,  
                      typeBoolean, sizeof( settable ), &settable );  
}
```



Example: Is Attribute Settable?

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeName,  
                  typeCFStringRef, NULL, sizeof( attribute ),  
                  NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    Boolean settable = true;  
    SetEventParameter( event,  
                      kEventParamAccessibleAttributeSettable,  
                      typeBoolean, sizeof( settable ), &settable );  
}
```



Example: Is Attribute Settable?

```
GetEventParameter( event,  
    kEventParamAccessibleAttributeName,  
    typeCFStringRef, NULL, sizeof( attribute ),  
    NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    Boolean settable = true;  
    SetEventParameter( event,  
        kEventParamAccessibleAttributeSettable,  
        typeBoolean, sizeof( settable ), &settable );  
}
```



Set Named Attribute

- kEventAccessibleSetNamedAttribute
 - UIElement being queried
 - Attribute name CFStringRef
 - Data to be set
- Extract the data and store it appropriately



Example: Set Named Attribute

```
GetEventParameter( event,  
    kEventParamAccessibleAttributeName,  
    typeCFStringRef, NULL, sizeof( attribute ),  
    NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    // ... extract text data and store it in the engine ...  
}
```



Example: Set Named Attribute

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeName,  
                  typeCFStringRef, NULL, sizeof( attribute ),  
                  NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    // ... extract text data and store it in the engine ...  
}
```



Example: Set Named Attribute

```
GetEventParameter( event,  
                  kEventParamAccessibleAttributeName,  
                  typeCFStringRef, NULL, sizeof( attribute ),  
                  NULL, &attribute );
```

```
if ( CFStringCompare( attribute, kAXTextAttribute )  
    == kCFCompareEqualTo )  
{  
    // ... extract text data and store it in the engine ...  
}
```



Example: Set Named Attribute

```
err = GetEventParameter( event,  
    kEventParamAccessibleAttributeValue,  
    typeCFTTypeRef, NULL, sizeof( data ),  
    NULL, &data );
```

```
if ( err != noErr )  
    return err;
```

```
if ( CFGetTypeID( data ) != CFStringGetTypeID() )  
    return eventNotHandledErr;
```

```
MyTextEngineSetTextFromString( engine,  
    (CFStringRef)data );
```



Example: Set Named Attribute

```
err = GetEventParameter( event,  
    kEventParamAccessibleAttributeValue,  
    typeCFTTypeRef, NULL, sizeof( data ),  
    NULL, &data );
```

```
if ( err != noErr )  
    return err;
```

```
if ( CFGetTypeID( data ) != CFStringGetTypeID() )  
    return eventNotHandledErr;
```

```
MyTextEngineSetTextFromString( engine,  
    (CFStringRef)data );
```



Example: Set Named Attribute

```
err = GetEventParameter( event,  
    kEventParamAccessibleAttributeValue,  
    typeCFTTypeRef, NULL, sizeof( data ),  
    NULL, &data );
```

```
if ( err != noErr )  
    return err;
```

```
if ( CFGetTypeID( data ) != CFStringGetTypeID() )  
    return eventNotHandledErr;
```

```
MyTextEngineSetTextFromString( engine,  
    (CFStringRef)data );
```



Example: Set Named Attribute

```
err = GetEventParameter( event,  
    kEventParamAccessibleAttributeValue,  
    typeCFTTypeRef, NULL, sizeof( data ),  
    NULL, &data );
```

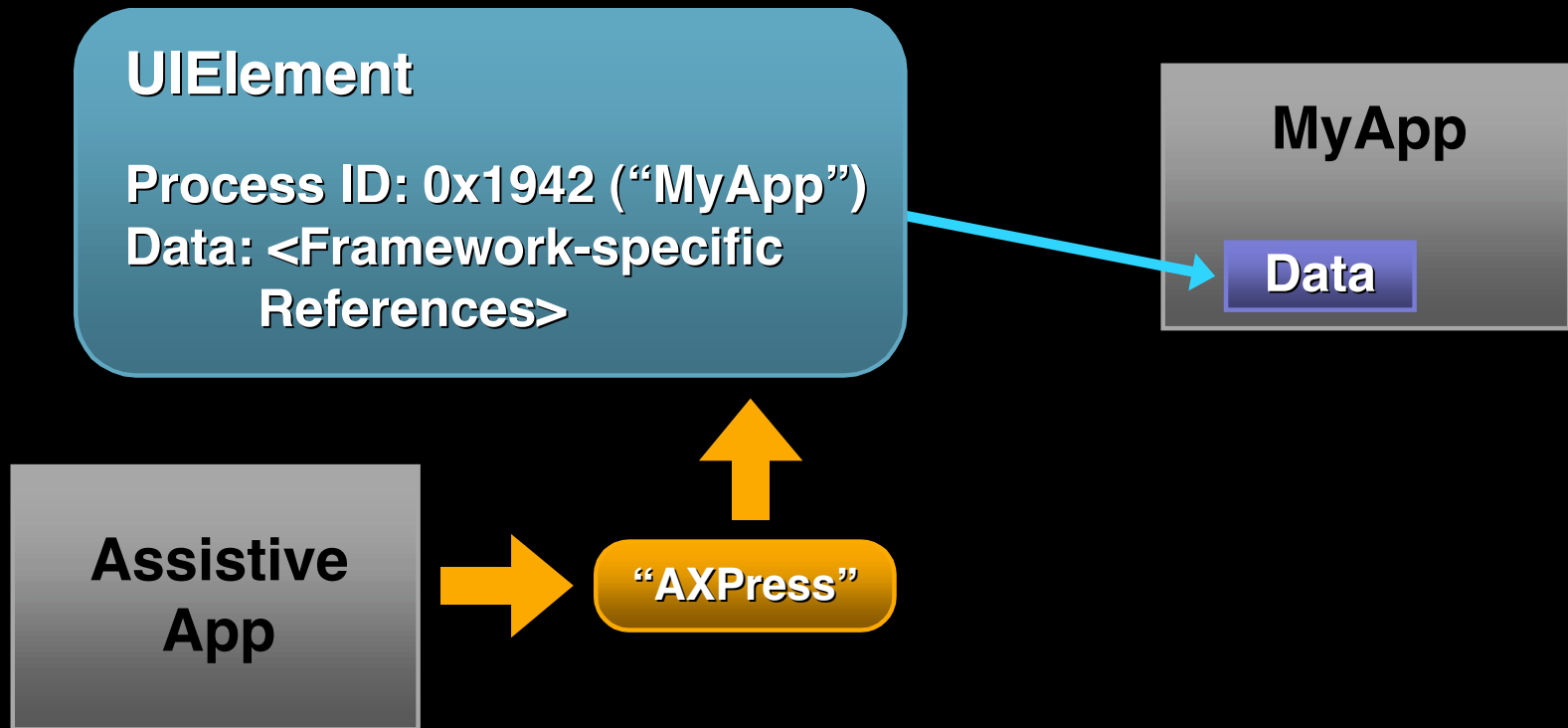
```
if ( err != noErr )  
    return err;
```

```
if ( CFGetTypeID( data ) != CFStringGetTypeID() )  
    return eventNotHandledErr;
```

```
MyTextEngineSetTextFromString( engine,  
    (CFStringRef)data );
```



UIElement Actions



UIElement Actions

- Things you can do to a UIElement
- Examples
 - Press a button
 - Pick a menu item
 - Close a menu



Get All Action Names

- `kEventAccessibleGetAllActionNames`
 - UIElement being queried
 - `CFArrayRef` to modify
- Add all supported action names to `CFArrayRef`
- Several standard action names
 - `<HIServices/AXConstants.h>`



Get Named Action Description

- kEventAccessibleGetNamedActionDescription
 - UIElement being queried
 - Action name CFStringRef
 - CFMutableStringRef to modify
- Alter the mutable string to contain a *localized* description of the action
 - Might be spoken by a screen reader



Perform Action

- `kEventAccessiblePerformNamedAction`
 - UIElement to perform the action
 - Action name `CFStringRef`
- Perform the action
 - Programmatically manipulate the interface



Example: Supporting AXPRESS

```
GetEventParameter( event,  
    kEventParamAccessibleObject, typeCTypeRef,  
    NULL, sizeof( buttonElement ), NULL,  
    &buttonElement );
```

```
buttonControl = AXUIElementGetHIObject(  
    buttonElement );
```

```
HViewSimulateClick( buttonControl,  
    kControlButtonPart, 0, NULL );
```



Example: Supporting AXPRESS

```
GetEventParameter( event,  
    kEventParamAccessibleObject, typeCTypeRef,  
    NULL, sizeof( buttonElement ), NULL,  
    &buttonElement );
```

```
buttonControl = AXUIElementGetHIObjec(  
    buttonElement );
```

```
HViewSimulateClick( buttonControl,  
    kControlButtonPart, 0, NULL );
```



Example: Supporting AXPRESS

```
GetEventParameter( event,  
    kEventParamAccessibleObject, typeCTypeRef,  
    NULL, sizeof( buttonElement ), NULL,  
    &buttonElement );
```

```
buttonControl = AXUIElementGetHIOBJECT(  
    buttonElement );
```

```
HViewSimulateClick( buttonControl,  
    kControlButtonPart, 0, NULL );
```

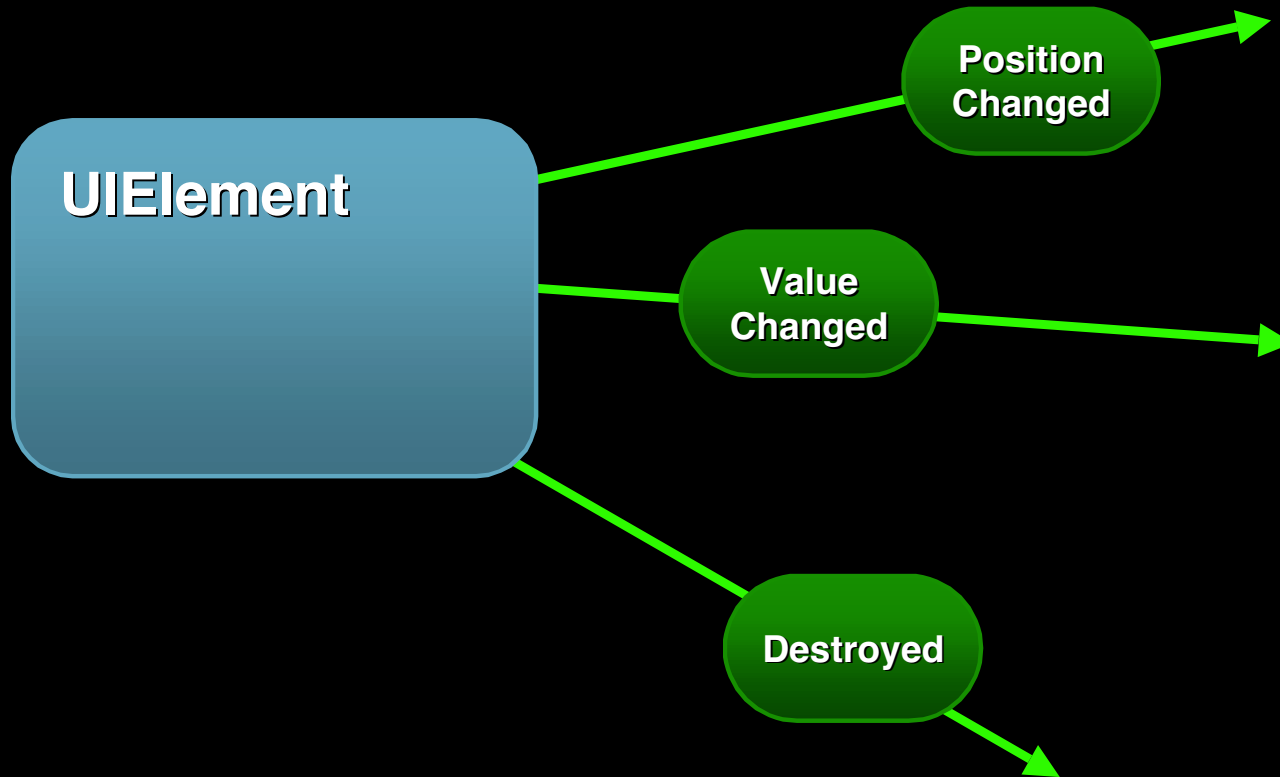


Sidebar: Use HICommands

- Required for accessibility actions to work!
- Put appropriate commands in:
 - Controls
 - Menu Items
- Remove post-HandleControlClick code
- Remove post-MenuEvent/MenuSelect code
- Handle kEventCommandProcess events



UIElement Notifications



UIElement Notifications

- Things that happen to a UIElement
- Assistive Apps listen to notifications
- Carbon already sends notifications for standard Controls, Windows, and Menus
- Many standard notifications
 - `<HIServices/AXNotification.h>`



Example: Posting Notifications

```
AXNotificationHIOBJECTNotify(  
    kAXValueChangedNotification,  
    myControlRef, 0 );
```



Text Editing Accessibility

- Text is *very* important!
 - Screen readers cannot work without it
- Make your text accessibility as rich as possible
- Report all the text in your implementation



Keyboard Navigation

- Most controls accept keyboard focus
- Hot keys allow:
 - Switching between windows
 - Focusing utility windows
 - Focusing toolbars
- Customization through Carbon Events
 - Details in Session 206



Not Done Yet . . .

- Not yet exported
 - AXNotificationHIOBJECTNotify
 - HIOBJECTSetAccessibilityIgnored
- Still defining Attributes, Actions, and Notifications
- Still implementing Accessibility functionality in system HIOBJECTs



Steps for Accessibility

- Use HICommands in your Menus and Controls
- In your custom content:
 - Install Accessibility Carbon Event Handlers
 - Send Notifications
 - Implement Keyboard Navigation



Documentation

Accessibility

- Making Your Application Accessible to Users With Disabilities

ADC Member Site > Download Software > “Jaguar” Mac OS X > Docs
connect.apple.com

- Accessibility Reference for Assistive Applications

“Jaguar” Mac OS X Developer Tools CD
[/Developer/Documentation/ReleaseNotes/AssistiveAPI.html](http://Developer/Documentation/ReleaseNotes/AssistiveAPI.html)



Roadmap

204 HIToolbox: An Architectural Overview: Get the details on HIObject

Hall 2
Wed., 9:00am

203 Migrating to Carbon Events:
Learn the Carbon Event basics

Hall 2
Tue., 5:00pm

009 Accessibility Overview:
Review the basic Accessibility concepts

Room A2
Thurs., 2:00pm



Who to Contact

Travis Brown

Graphics and Imaging Evangelist

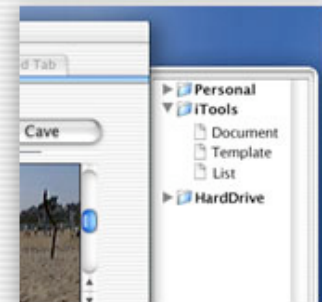
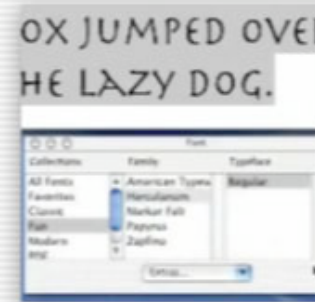
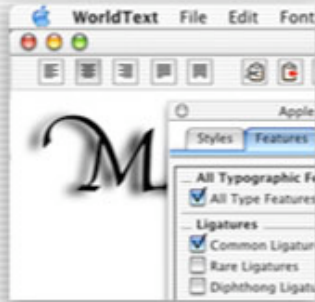
travis@apple.com

<http://developer.apple.com/wwdc2002/urls.html>





Q&A



Travis Brown
Graphics and Imaging Evangelist
Worldwide Developer Relations

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**

 **WWDC2002**